WWW.SCIAM-MAGAZINE.NAROD.RU

Путешествие с роботами в Роботрополе и овладение навыками конструирования электронных микросхем

А. К. ДЬЮДНИ

Подобно сталкивающимся галактикам, компьютерные игры и обучение с помощью компьютеров, постепенно сближаясь, кажется, готовы слиться воедино. Предвестником этого события можно считать появление в прошлом году компьютерной игры под названием «Одиссея роботов», которая одновременно с развлечением служит целям обучения в области конструирования электронных схем.

Уже создано немало программ, обучающих на основе игрового подхода. Однако ни в одной из них, насколько мне известно, не было достигнуто той степени интенсивности обучения и увлекательности, которая присуща игре «Одиссея роботов». Многие компьютерные игры являются результатом эволюции на пути от первобытного хаоса видеоигр к более упорядоченным и более интеллектуальным формам. По своей образовательной ценности ни одна из этих игр не может сравниться с «Одиссеей роботов».

«Одиссея роботов» ставит игрока в безвыходное положение: чтобы выбраться из Роботрополя — фантастического города, населенного исключительно роботами, играющий вынужден заняться конструированием логических схем. Большинство автоматов — обитателей города относится к людям враждебно. Когда игрок (представленный на экране дисплея в виде маленького гомункулуса) пытается выйти из лабиринта или заполучить нужный ему предмет, робот-сторож хватает гомункулуса и бесцеремонно вышвыривает его из помещения. Однако три робота оказываются полезными компаньонами в путешествии. Схемная конструкция роботов-компаньонов, которых роботы-сторожа не замечают, такова, что позволяет избегать столкновений с другими автоматами — обитателями Роботрополя, успешно выбираться из лабиринтов и завладевать такими предметами, как жетоны на проезд в метро и кристаллы — аккумуляторы энергии.

Может ли играющий на опыте проделанного путешествия сконструировать компьютер? Наверное, нет, однако упражнения окажутся весьма полезными. Игра начинается в помещении, называемом комнатой роботов. Здесь гомункулус, управляемый с клавиатуры, встречается с тремя роботами-компаньонами, весело пляшущими в прямоугольной области. Каждый робот ведет себя по-своему. Один скользит по комнате, прижимаясь к стенам. Другой скачет в горизонтальной плоскости взад и вперед. Третий методично обследует правую стену, как будто отыскивая выход. Играющий начинает с того, что набирает на клавиатуре букву R, и затем нажимает на клавишу, соответствующую пробелу. Это действие производит сюрреалистический эффект — гомункулус превращается в устройство, управляющее на расстоянии. Оно немедленно приказывает роботам остановиться. Теперь играющий может управлять гомункулусом. По команде последний, например, может забраться на любого робота. Когда он это делает, картинка на экране меняется: вместо происходящего в комнате действия появляется схематическое изображение внутреннего устройства робота. Гумункулус уменьшается в десять раз. Он может оседлать робота, пытаясь, например, найти выход из лабиринта или в какой-нибудь другой ситуации, когда манера передвижения данного робота оказывается подходящей.

Каждый робот-компаньон обладает сенсорными устройствами, благодаря которым он ориентируется в окружающей обстановке и может целенаправленно передвигаться. С помощью бамперов он ощущает контакт со стеной. У робота есть движки, толкающие его вверх, вниз, влево и вправо. Автоматическая рука с захватом может брать и высвобождать предметы. Связь с двумя другими роботами-компаньонами осуществляется при помощи антенны. Внутри каждого робота это оборудование представляется в виде соединительных линий, которые можно «припаять» к тому или иному участку логической схемы робота. Эти соединительные линии могут быть двух типов: входные, через которые передается информация о состоянии соответствующего сенсорного устройства (бампера, механической руки, антенны), и выходные, посредством которых команды от внутренней логической схемы передаются устройствам-эффекторам (толкающим движкам, механической руке, антенне). Таким образом, механическая рука и антенна имеют как входные, так и выходные соединительные линии. Выходная линия механической руки приводит ее в действие, а входная линия позволяет установить, держит ли рука в данный момент какой-нибудь предмет.

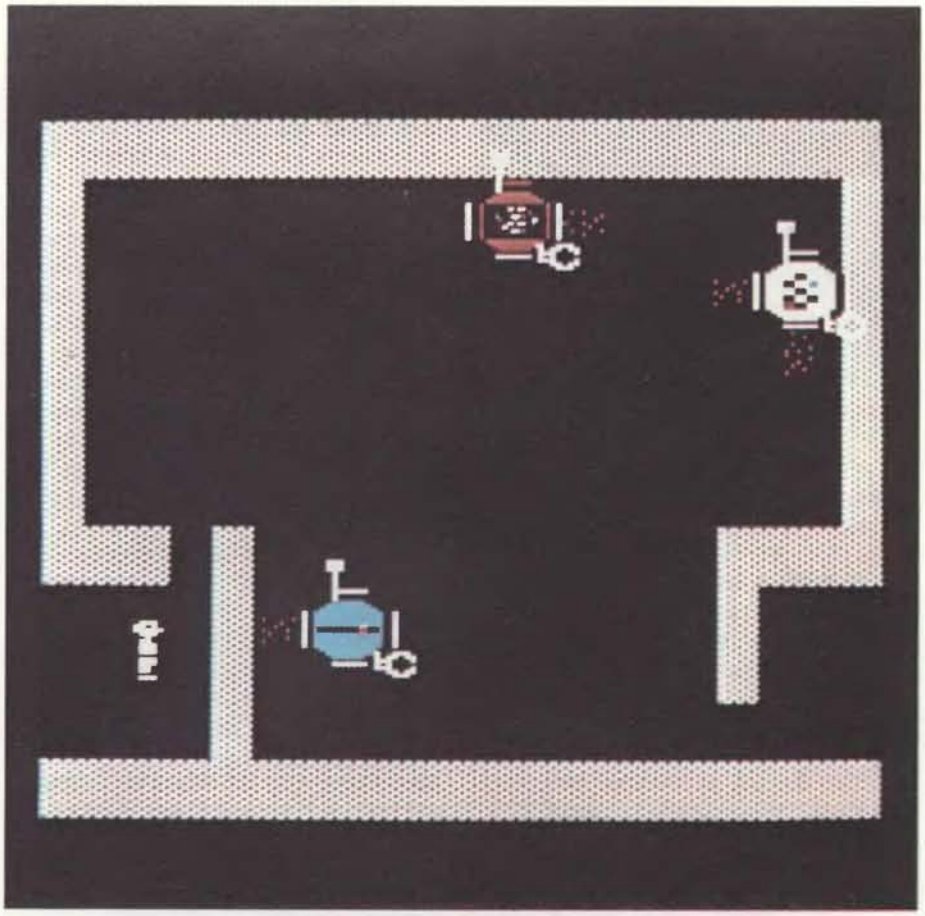
Посмотрим, как работают эти электронные устройства, наблюдая за поведением самого простого из трех роботов-компаньонов. Этот робот по имени Пинг-понг управляется единственным триггерным элементом, подсоединенным к горизонтальным бамперам и движкам. В «Одиссее роботов» триггерный элемент представляет собой двустороннюю ячейку памяти, в которой одна сторона содержит единицу, в то время как другая — нуль. Когда по входной линии, подсоединенной к нулевой стороне, приходит сигнал, пусть даже длящийся одно мгновение, эта сторона переключается с нуля на единицу, а другая сторона — с единицы на нуль. При отсутствии изменений на входах триггер сохраняет свое состояние. Он как бы запоминает его. Чтобы реализовать поведение Пинг-понга, не требуется ничего, кроме описанного простого триггера. Выходная линия с одной стороны триггера соединена с правым движком робота, а входная линия с той же стороны триггера соединена с правым бампером. Если на этой стороне триггера содержится нуль и робот коснется стены справа, то по входной линии сигнал от бампера передается триггеру. Триггер немедленно меняет свое состояние — там, где был нуль, теперь единица. В результате включается правый движок и робот начинает двигаться влево. Назовем эту стадию «пинг». Стадия «понг» реализуется аналогичным образом путем соединения противоположной стороны триггера с левым бампером и левым движком.

Чтобы реализовать более сложное поведение, требуется полный стандартный набор логических элементов. В этот набор помимо триггерного запоминающего элемента входят также элементы, принимающие решение и называемые вентилями типа ИЛИ, И и НЕ. Последний называется также инвертером. На выходе вентиля ИЛИ всегда единица, когда по крайней мере на одном из его входов имеется единица. На выходе вентиля И будет единица, когда на всех его входах единица. На выходе инвертера единица будет тогда, когда на его входе нуль.

Чтобы покинуть комнату роботов,

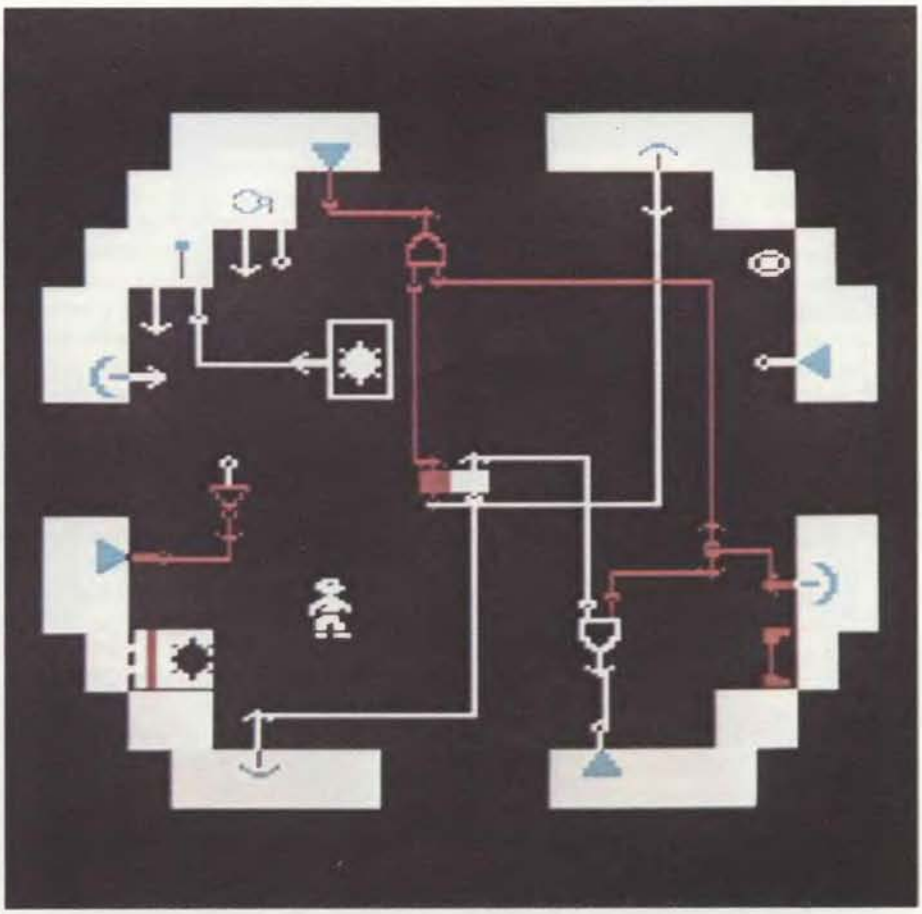
WWW.SCIAM-MAGAZINE.NAROD.RU

играющий должен извлечь ключ, находящийся внутри одного из роботов, и открыть дверь, ведущую к месту, называемому городской свалкой. Чтобы подобрать какой-нибудь предмет, нужно «наехать» на него гомункулусом и нажать клавишу пробела. Об успехе действия свидетельствует звуковой сигнал. После этого предмет движется вместе с гомункулусом. Без этого средства игра «Одиссея роботов» была бы невозможна: в своих путешествиях по пяти уровням игры гомункулус должен нести с собой трех роботов-компаньонов. Да, но как он может нести с собой роботов, если по условиям игры нести можно только один предмет? Поскольку любой предмет можно поместить внутрь робота и так как робот тоже представляет собой предмет, играющий быстро догадывается, что одного робота можно поместить внутрь другого. Этот прием действует. На самом деле гомункулус может нести с собой робота, внутри которого находится второй робот, а в этом последнем заключен третий. Благодаря такому приему можно сэкономить место внутри внешнего робота. Это необходимо, потому что по мере развития игры в нем накапливаются разнообразные предметы, такие, как жетоны на проезд, сенсоры, кристаллы-аккумуляторы и всевозможные логические элементы.



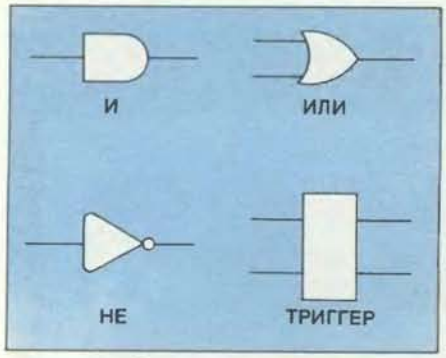
Вход в Роботрополе и три робота-компаньона

Моя собственная партия игры в «Одиссею роботов» началась, как мне показалось, с бесконечного путешествия по помещениям, прилегающим к городской свалке. В центральном помещении я набрел на комнату, ревностно охраняемую одним из самых неприятных обитателей Роботрополя. В дальней части комнаты у стены лежал кристалл—аккумулятор энергии. Стремясь завладеть им, я извлек робота, исследующего стены, из Пинг-понга и привел его в движение. Незамеченный сторожем робот заскользил по стене. Наткнувшись на кристалл, робот захватил его механической рукой, издав при этом звуковой сигнал, и продолжил свое движение вдоль стены, теперь уже по направлению к выходу. Здесь я выключил робота, ходящего вдоль стен, и попытался извлечь кристалл-аккумулятор из его механической руки. Робот почему-то не выпускал его. Отложив решение этой задачи на более позднее время, я снова поместил робота, ходящего вдоль стен, внутрь Пинг-понга и хотел уже было продолжить свою одиссею, когда услышал какое-то скрежетание, а затем постепенно слабеющий звук, сигнализирующий о том, что питающая батарея истощилась. Оказывается, робот по имени Вампир только что запустил свой электрический хобот в Пинг-



Бамперы, толкающие движки, механическая рука, антенна, соединительные линии и электронные схемы правостороннего робота

WWW.SCIAM-MAGAZINE.NAROD.RU



Логические элементы для игры «Одиссея роботов»

понга и высосал из него половину энергии. В Роботрополе нужно всегда быть начеку.

Поблуждав некоторое время по городу и подобрав по пути кое-какие полезные с виду предметы, я вышел к проходу с надписью «Есть ли у вас все необходимое?». И тут я вспомнил, что позабыл забрать магнит из соседнего с городской свалкой помещения, так как боялся встретить Вампира, который, по-видимому, патрулировал в районе свалки. Мой гомункулус миновал надпись, и перед ним открылся лабиринт городской свалки. Он представлял собой вертикальные стены, то поднимающиеся снизу вверх, то свисающие сверху вниз (см. рисунок внизу). Лабиринт охранялся еще одним роботом, встреча с которым не предвещала ничего хорошего.

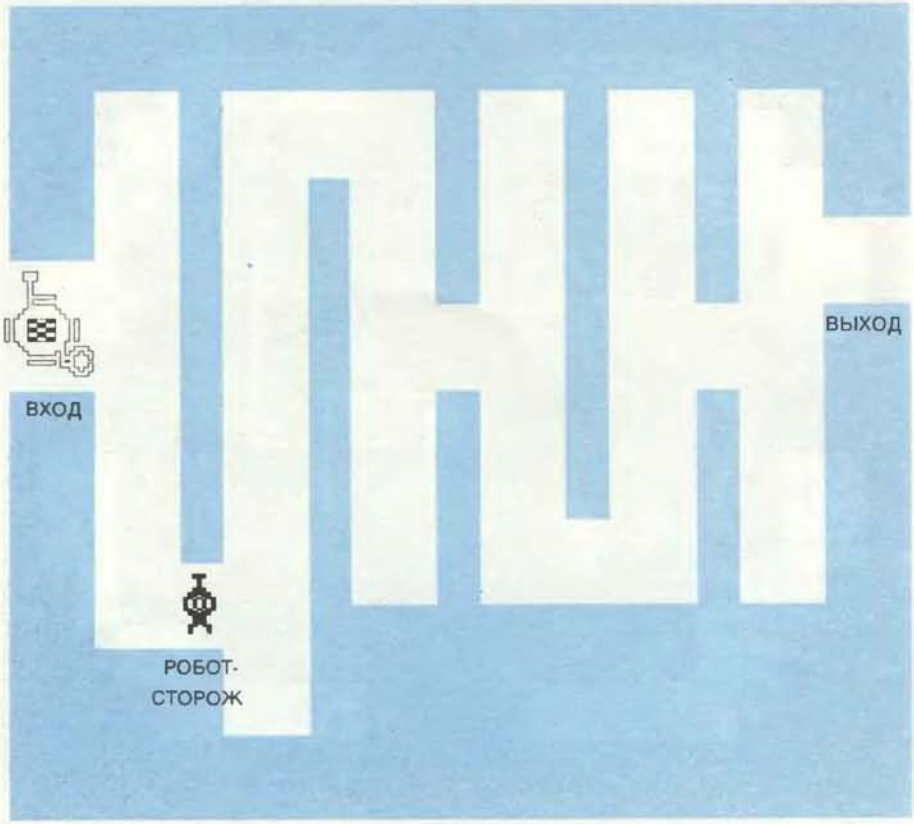
Один из моих роботов-компаньонов, как мне показалось, был создан для подобных ситуаций. Когда включается робот, которого я называю правосторонним, он всегда движется вправо. Если его дальнейшему движению по правой стороне мешает стена, правосторонний робот начинает скользить вдоль нее, пока не упрется в горизонтальную стену, где он сразу меняет направление движения. Поведение робота становится понятным, если посмотреть на его внутреннюю логическую схему (см. нижний рисунок на с. 95). Во-первых, левый движок всегда включен, поскольку линия, управляющая им, проходит через инвертер, на который не подается сигнал (и так как на входе инвертера нуль, то на его выходе единица). Управление вертикальными движками осуществляется триггерным элементом, входы и выходы которого аналогичны схеме, управляющей Пинг-понгом. Отличие, однако, состоит в том, что выходные линии, подсоединенные к вертикальным движкам, проходят через вентили И. В качестве второго входа для каждого вентиля И выступает линия, соединенная с правым бампером. Это означает, что команда включить верхний или нижний движок не будет выполнена, если правый бампер не находится в контакте с каким-либо препятствием.

Я извлек правостороннего робота из робота, которого я нес с собой, и посадил на него своего гомункулуса. Поместив гомункулуса непосредственно на место символа, имеющего форму глаза, я получил возможность перископического обозрения окружающей обстановки вне правостороннего робота. Его внутренний вид сменился на экране дисплея видом лабиринта. В левой части картинки я увидел правостороннего робота с торчащим из него перископом. Правда, было не совсем понятно, каким образом сцена могла включать сам прибор, из которого ведется наблюдение, однако я не стал об этом задумываться.

Гомункулус, устроившийся у перископического глаза, мог теперь выполнять функции управления. Я нажал букву *R*, а затем пробел. Правосторонний робот вместе с сидящим в нем гомункулусом заскользил по лабиринту и уперся в стену. Пройдя вверх вдоль стены, мы дошли до тупика, отскочили вниз и добрались до прохода в стене. Здесь под нависающей сверху стеной мы увидели останки неудачливых гомункулусов, не сумевших найти выход из лабиринта. Далее мы проскользнули мимо робота-сторожа, который, казалось, не замечал, что вместе с правосторонним роботом движется человеческое существо. Миновав проход, мы встретились с еще одной вертикальной стеной и поползли вверх. Когда мы наконец выбрались из лабиринта, я понял, что совершил ужасную ошибку, оставив двух других роботов-компаньонов позади. Страх встречи с Вампиром сбил меня с толку. Каким образом можно было вернуться за моими роботами? Нужно было переделать правостороннего робота. Возможно, читатели, поразмыслив над его устройством, сами догадываются, как это сделать.

Чтобы перепаять какую-нубудь схему робота, нужно нажать букву *S* — и гомункулус превращается в паяльник. В этом режиме клавиша пробела используется, чтобы разорвать старые соединения и создать новые. Если вам не хватает какой-нибудь детали, нужно нажать на букву *T* — и на экране появляется набор деталей. Гомункулус может погрузиться в него и взять нужный элемент.

Когда я наконец собрал всех трех роботов-компаньонов у выхода из лабиринта, моя одиссея продолжалась. Теперь на нашем пути была запертая комната. Воспользовавшись ключом, я открыл ее и, войдя, обнаружил, что у нее не было другого выхода. Куда же идти? На стене я увидел зловещую



Лабиринт городской свалки, выход из которого должен найти правосторонний робот

WWW.SCIAM-MAGAZINE.NAROD.RU

надпись: «Приготовьтесь, вы будете переведены в другое измерение!» Посреди комнаты в неподвижном прямоугольнике плясал двойник моего гомункулуса.

В ходе игры можно в любой момент запомнить ее состояние, записав его на свободный диск. При этом игра как бы спасается. Если следующий ход обернется катастрофой, то игру можно переиграть, начиная с запомненного положения. Эта возможность присуща многим приключенческим играм, реализованным на компьютерах. Надеемся, что у молодых людей, увлекающихся этими играми, не сложится впечатление, что и в реальной жизни, полной приключений, тоже можно вернуться к какому-то моменту в прошлом, чтобы потом все сделать по-другому.

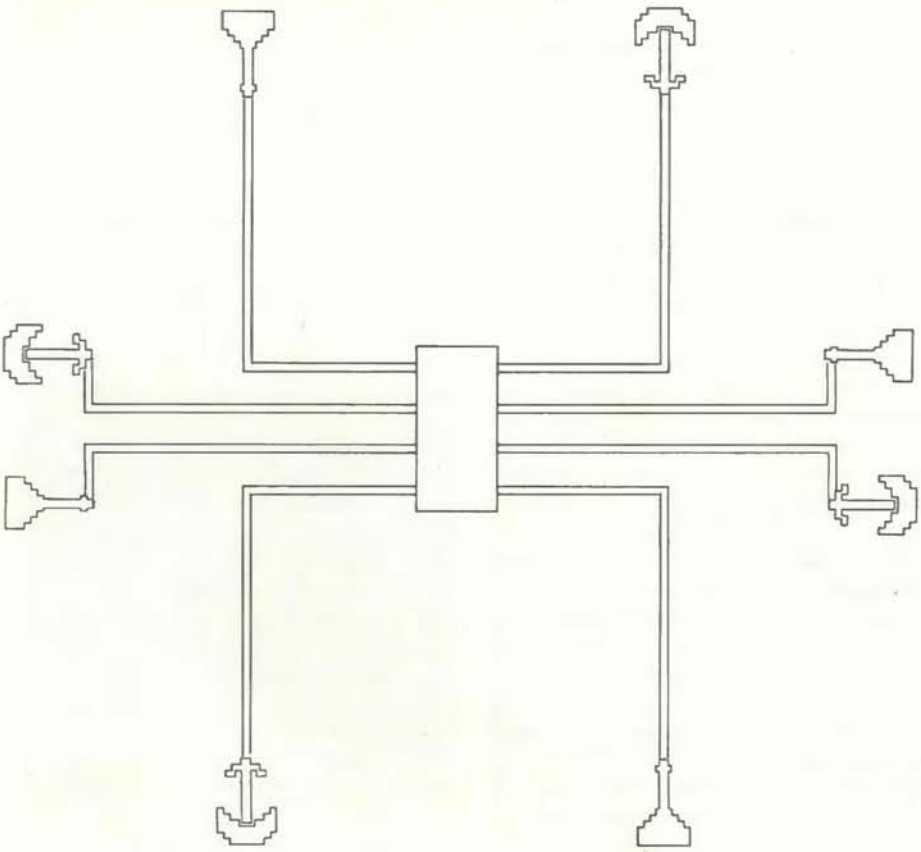
Таким образом я спас игру и, наехав своим гомункулусом на его двойника, нажал на клавишу пробела. Казалось, все взорвалось в фейерверке искр, и мой переход в новое измерение сопровождался каким-то странным звуком. На самом деле я очутился в другом помещении на втором уровне Роботрополя, но... без своих компаньонов. Очевидно, я забыл из взять с собой. Воспользовавшись запомненным положением, теперь можно было переиграть фазу, предшествовавшую переходу в другое измерение, приняв меры к тому, чтобы не потерять своих роботов.

Я не буду описывать своих дальнейших приключений в Роботрополе, чтобы не лишать удовольствия читателей, которые пожелают сами сыграть в эту игру.

Помимо самой игры покупатели «Одиссеи роботов» получают записанные на диске сведения об анатомии роботов, о том, как пользоваться набором деталей, а также руководство по конструированию электронных логических схем. Конструирование происходит в помещении, называемом лабораторией новой техники.

В любой момент в ходе игры играющий может перейти в лабораторию и испытать новые логические схемы на роботах, которые там находятся. По мере того как схемы увеличиваются в размерах и усложняются, становится целесообразным реализовывать их в виде интегральных схем (т.е. схем, смонтированных в одном кристалле). Это делается в специально отведенной комнате лаборатории.

Один из роботов-компаньонов, а именно робот, двигающийся вдоль стены, был уже снабжен такой интегральной схемой. Входные и выходные соединительные линии этой интегральной схемы изображены на рисунке, приведенном вверху. Читателям, наверное, будет интересно самостоятельно сконструировать интегральную схему для управления роботом, ходящим вдоль стены. При этом разрешается пользоваться только триггерами и логическими элементами, о которых говорилось выше. Я с удовольствием рассмотрю присланные читателями схемы, однако некоторым читателям, возможно, захочется самим посмотреть на свои конструкции в действии.



Бамперы, толкающие движки и соединительные линии робота, ходящего вдоль стены. Какая электронная логическая схема управляет им?

Игра «Одиссея роботов» была изобретена и реализована в виде компьютерной программы М. Уоллесом, когда он был студентом Станфордского университета, и Л. Гриммом, сотрудником фирмы Learning Company. У этой игры в коллекции фирмы Learning Company были предшественники. В таких играх, как «Головоломки Гертруды» и «Сапоги Рокки», от играющего также требуется умение конструировать логические схемы для решения задач. У. Робиннет, в прошлом сотрудник фирмы, придумал для всех этих игр интересную пространственную среду, в которой должны происходить игровые действия. Например, помещения и проходы Роботрополя конечны, но они сменяют друг друга до бесконечности, пока играющий не поймет, что, покидая комнату через правый выход, он может очутиться у левого входа какой-нибудь комнаты, в которой он уже побывал.

Конечно, логические схемы, конструируемые для игры «Одиссея роботов», не сравнить по сложности со схемами для современного компьютера. Тем не менее, пользуясь компонентами, имеющимися в распоряжении играющего, можно построить простейший компьютер. Игроки, усвоившие правила игры и хорошо пользующиеся ими, возможно, захотят узнать, можно ли создать компьютер, для того чтобы встроить его в робот. Такой компьютер мог бы быть программируемым путем ввода последовательности нулей и единиц перед тем, как робот начнет действовать. Такая последовательность играла бы роль ультрапростого языка, а подпоследовательности из нулей и единиц представляли бы, например, такие команды, как «Двигаться вправо до стены» или «Привести в действие механический захват».

Процесс конструирования логической схемы современного компьютера сводится к размещению огромного числа полупроводниковых транзисторов и других электронных компонентов в виде большой плоской схемы, хранящейся в памяти ЭВМ. После проверки и отладки с помощью моделирования эта схема переносится на фотопленку и затем вытравляется в

WWW.SCIAM-MAGAZINE.NAROD.RU

кремниевом кристалле. На первый взгляд этот очень сложный процесс имеет мало общего с вентилями и триггерами, используемыми в игре «Одиссея роботов», и даже с технологией, описываемой в большинстве университетских учебников по основам конструирования логических схем. Однако между тем и другим все же есть немало общего. Как объяснил мне несколько лет назад К. Клейр, теперь работающий в фирме CTX International в г. Саннивейл (шт. Калифорния), учащиеся, занимаясь построением стандартных логических схем, развивают в себе способность к абстрактному мышлению, которое необходимо будущему инженеру — конструктору электронных устройств. При разработке той или иной электронной схемы сначала нужно представить задачу на абстрактном уровне и попытаться решить ее, оперируя только вентилями И, ИЛИ, НЕ и триггерными элементами. Навыки абстрактного мышления помогают затем и при работе на более сложном уровне реальных полупроводниковых приборов, таких, как полевые транзисторы и другие современные электронные компоненты, используемые в различных логических устройствах.

По словам П. Эшкина, одного из ведущих специалистов-разработчиков фирмы Apple, производящей персональные компьютеры, игра «Одиссея роботов» является в этом смысле весьма полезной. Эшкин говорит, что благодаря этой игре учащийся развивает в себе два навыка, необходимых хорошему инженеру — конструктору ЭВМ: умение четко осознать поставленную задачу и способность представить решение в виде логических схем любого типа. Например, когда я в последний раз играл в «Одиссею роботов», передо мной возникла задача завладеть жетоном, лежащим посредине охраняемой комнаты. Было ясно, что это дело следует поручить одному из роботов-компаньонов. Однако каким образом он мог осуществить эту операцию? Робот, ходящий вдоль стены, здесь был бесполезен, поскольку он никогда не может приблизиться к жетону настолько, чтобы захватить его механической рукой. В моем распоряжении был специальный датчик, обнаруживающий жетоны и имеющий выходы, позволяющие определить направление на жетон. С помощью этого прибора робот, несомненно, найдет жетон, но, к сожалению, тот же прибор не позволит роботу покинуть то место, где был найден жетон. Во всяком случае, проанализировав ситуацию, я выделил две отдельные задачи: построить схему, отыскивающую жетоны, и найти способ запрограммировать робота так, чтобы он мог выбраться из комнаты. Хотя некоторые задачи выглядят сначала несколько расплывчато, Эшкин отмечает, что их маленькие, «байтового размера» подзадачи поддаются решению. При таком разбиении постановка задачи становится более четкой. Процесс постановки задачи и решение составляющих ее подзадач на уровне логических схем — две важные фазы разработки устройства.

То, что играющий обретает способность самостоятельно конструировать логические схемы, имеет также важное психологическое значение. По мере того как играющий приобретает опыт, перепаивая схемы роботов-компаньонов, у него развивается чувство уверенности в своих силах, а уверенность — это отличное противоядие от ощущения холодной неприступности, которое порой внушают нам компьютеры. Обучившись оперировать стандартными логическими схемами в игре «Одиссея роботов», играющий подходит к пониманию аналогичных взаимоотношений, существующих между основными логическими элементами компьютера и его функционированием. Играющий чувствует, что нет никаких непреодолимых препятствий на пути более глубокого исследования. Ощущение загадки, таящейся в черном ящике, пропадает. Человек начинает чувствовать себя полным хозяином положения.

Однако это ощущение всесилия, развивающееся на определенной стадии у начинающих, может затем снова быть утраченным, как утверждает Ш. Тёркл, автор книги «The Second Self» («Второе Я»). Ее книга посвящена тому, как развиваются взаимоотношения между людьми и компьютерами. Несколько страшноватыми выглядят некоторые истории о тех, кто начинает рассматривать самого себя как машиноподобное существо. Мне представляется, что подобная точка зрения может развиться на любой стадии овладения теорией и практикой вычислительной техники. Я знаком со специалистами в области искусственного интеллекта, которые также находят в себе нечто общее с машиной, но, правда, в более тонком понимании этого слова. В каком-то смысле, может быть, и правда, что все мы машины, однако такая точка зрения мало помогает нам в вопросах, касающихся нашего поведения, и может даже способствовать тому, что человек примет такую линию поведения, которая приведет к самым неприятным последствиям. Возможно, именно поэтому, образно говоря, нам следует научиться убегать из Роботрополя.

ПИСЬМА читателей, откликнувшихся на статью «Пять несложных программ для различных задач моделирования с помощью случайных чисел» («В мире науки», 1985, № 6), свидетельсвуют о том, что полученные ими результаты действительно приводят к гистограмме, очень близкой к отрицательному экспоненциальному распределению. Сотням читаталей удалось написать первую пьесу — программу для приближенного вычисления числа π. И лишь несколько читателей справились с пятой пьесой — программой, моделирующей поведение очередей.

В программе вычисления числа π моделируемые пушки стреляют по моделируемому полю, окружающему моделируемый пруд. Я приношу свои извинения тысячам соседей, которые не смогли заснуть под раскаты пушечной стрельбы, продолжавшейся ночи напролет.

Написав программу, вычисляющую подобным образом приближенное значение числа π, очевидно, трудно удержаться от соблазна произвести больше чем 1000 выстрелов. Чемпионом в этом упражнении следует признать Дж. Саймонса из Медицинского училища Гарвардского университета. Продолжая выполнение программы в течение целой недели на мощной ЭВМ, Саймонс сумел произвести более 4 млрд. выстрелов по пруду. Полученное им значение числа π равно 3,14157. К сожалению, пятый разряд после запятой получился равным 7, а не 9, как должно быть.

На данный момент лишь 134 читателя представили свои оценки числа π, полученные в результате цикла из 1000 выстрелов. В тех случаях, когда читатели представляли результаты нескольких таких циклов, я выбирал только первый результат для того, чтобы получить среднее значение. При этом средний результат оказался равным 3,14137. Первые три знака после запятой правильны. В четвертом разряде числа π должно быть не 3, а 5.

Между прочим, занимаясь этими подсчетами, я ни разу не заглянул в справочник, чтобы вспомнить точное значение числа π. Вместо этого я воспользовался мнемоническим правилом: «How I wish I could enumerate pi easily» («Как я хочу π числа выражение не забыть»). Выпишите последовательно число букв в каждом слове этой фразы и вы получите число π. Сумеет ли кто-нибудь предложить продолжение для этого мнемонического правила?

Обе приведенные оценки числа π оказываются несколько меньше точного значения. Этот сдвиг в меньшую сторону был предсказан Дж. Файнапом из Энн-Арбора (шт. Мичиган), по

WWW.SCIAM-MAGAZINE.NARODU.RU

словам которого большинство датчиков случайных чисел имеет небольшое отклонение в сторону нуля. Это отклонение повлияло в том же направлении (в сторону уменьшения) и на программу подсчета числа π.

Дж. Мак-Кин, статистик из Университета Западного Мичигана в Каламазу (шт. Мичиган), указывает, что при подобных подсчетах можно установить 95%-ный доверительный интервал вокруг полученной оценки. При этом предполагается, что полученное приближенное значение находится посередине числового интервала, составляющего доверительный интервал, а сам интервал определяется таким образом, что точное значение с вероятностью 95% находится в какой-то точке интервала. Размер последнего меняется обратно пропорционально квадратному корню из n — количества произведенных выстрелов. Отсюда следует, что для увеличения точности оценки, скажем, в 10 раз, или на один десятичный разряд, число n необходимо увеличить в 100 раз. Таким образом, приходится идти на большие вычислительные затраты, чтобы получить даже небольшое увеличение точности. Этим объясняется неудовольствие, выраженное несколькими читателями.

Все, кто написал программу ZOMBIE, без труда получили диаграммы, аналогичные экспоненциальной кривой, изображенной на рисунке, приведенном в июньском номере журнала «В мире науки» 1985, на с. 91.

Программа GALTON заинтриговала Э. Кудала из Парк-Риджа (шт. Иллинойс). Моделируемые программой шарики, скатываясь через треугольный массив колышков, образовали искомое распределение у основания доски Гальтона, а именно биномиальное распределение. Биномиальное распределение дискретно и имеет колоколообразную форму, имитируя своего собрата — нормальное распределение.

Двое читателей одновременно сделали одно и то же изобретение. Они решили обобщить программу GALTON для случая трехмерного массива колышков. У основания своих моделируемых досок Дж. Матушевич из Колумбуса (шт. Небраска) и Дж. Ньюджент из Пеории (шт. Иллинойс) получили гладкое возвышение из шариков. Может быть, читатели догадаются, какую форму имеет это возвышение?

Некоторые читатели, попробовавшие написать программу VOTERS, были разочарованы полученными результатами. В одном случае программа проработала всю ночь, однако развязка так и не наступила. Единственное, что я могу порекомендовать экспериментаторам, — это запастись терпением и продолжать работу с программой. Мне, например, потребовалась бóльшая часть суток. Следует также внимательно следить за развитием игры.

Наконец, только двое читателей попытались создать программу QWING, моделирующую очередь клиентов в банке с единственным окошком для обслуживания. В. Ауэрбах из Филадельфии (шт. Пенсильвания) написал свою оригинальную версию программы с разбиением по времени, благодаря чему он наблюдал, как очередь медленно растет, несмотря на многократные кратковременные уменьшения. Дж. Гудман из Куинси (шт. Массачусетс) внес еще более значительные изменения в программу QWING: он заменил систему моделирования очереди двумя дверями для человечков в одной-единственной стене. Читателям, возможно, будет интересно попытаться понять, как работает программа Гудмана, зная о ней только то, что было сказано выше.

Я уже отмечал, что ожидаемая длина очереди не является ни конечной, ни бесконечной, если средние длительности обслуживания одного клиента и интервала между прибытиями клиентов совпадают. Такая очередь вначале растет, но затем уменьшается до нуля. Далее она снова начинает расти, достигая даже еще больших размеров по сравнению с предыдущим разом, но затем опять сокращается до нуля. Длина очереди колеблется таким образом со все возрастающей амплитудой. Объяснение этого явления состоит в том, что, когда в очереди нет клиентов, потенциальное время их обслуживания навсегда теряется.

Многие читатели подозревают, что их генераторы случайных чисел содержат ошибки. Р. Гай из Оклендского университета в Рочестере (шт. Мичиган), показал, что для этих подозрений есть основания, по крайней мере в отношении генератора случайных чисел RND, используемом в языке Бейсик персонального компьютера фирмы IBM. Следующая программа, выполняющаяся только в графическом режиме, представляет собой наглядное свидетельство:

```
10 SCREEN 2
20 X = RND* 640
30 Y = RND* 200
40 PSET (X, Y)
50 GO TO 20
```

Если бы генератор RND работал безупречно, то точки на экране были бы разбросаны случайным образом, на самом деле на экране появляются кольцеобразные сгущения точек. Несомненно, аналогичное явление можно наблюдать и у других компьютеров.

Программа «червяк», текст которой был приведен в майском номере журнала «В мире науки» 1985, в рубрике «Занимательный компьютер», предназначенная для персонального компьютера APPLE II, видимо, сама явилась жертвой небольшой типографской инфекции. Второе число в команде DATA под номером 100 должно быть равно 255. В неисправленном варианте «червяк» проползает лишь часть пути по памяти и внезапно погибает.

Издательство **МИР** *предлагает:*

Д. Сахал

ТЕХНИЧЕСКИЙ ПРОГРЕСС: КОНЦЕПЦИИ, МОДЕЛИ, ОЦЕНКИ

Перевод с английского

Автор рассматривает различные подходы, позволяющие с помощью статистических методов оценить влияние технического прогресса на экономический рост. Анализируются социально-экономические последствия технического прогресса, вскрывается его внутренний механизм, рассматриваются взаимоотношения науки и техники. Книгу отличают глубина, всесторонность охвата рассматриваемых проблем, объективность; критически осмысливается современное состояние исследований.

Книга Д. Сахала — значительный шаг вперед в понимании технологии как самостоятельного фактора функционирования производственных и социальных систем, обладающего собственной динамикой.

1985, 25 л., с иллюстрациями.

Цена 2 р. 80 к.

ИЗДАТЕЛЬСТВО МИР